**ReadMe File - Auto SQLI Attack**

Contents

[How to run 3](#_Toc121289091)

[Move into VM 3](#_Toc121289092)

[Check User Settings 3](#_Toc121289093)

[Use Sudo 3](#_Toc121289094)

[Python 3 3](#_Toc121289095)

[Breakdown of Files 3](#_Toc121289096)

[ExtModulesSetUp.py 3](#_Toc121289097)

[KaliSetUp.py 4](#_Toc121289098)

[LinkSetup.py 4](#_Toc121289099)

[UserSetUp.py 4](#_Toc121289100)

[SqliStatementGen 4](#_Toc121289101)

[**most\_common\_passwords.csv** 4](#_Toc121289102)

[**most\_common\_passwords\_2.csv** 4](#_Toc121289103)

[Poss\_SQLI\_Injections.csv 4](#_Toc121289104)

[**Poss\_COMS\_Injections.csv** 4](#_Toc121289105)

[BrowserSetUp.py 4](#_Toc121289106)

[BrowserSetUp.py 4](#_Toc121289107)

[Pcap\_Capture.py 5](#_Toc121289108)

[PassCracker.py 5](#_Toc121289109)

[NameGenerator.py 5](#_Toc121289110)

[AutoSQLI.py 5](#_Toc121289111)

[**AutoComsI.py** 5](#_Toc121289112)

[**AutoBruteForce.py** 5](#_Toc121289113)

[/Drivers 5](#_Toc121289114)

[/SQLIATTACKS 5](#_Toc121289115)

[**/COMSIATTACKS** 5](#_Toc121289116)

[**/BRUTEFORCEATTACKS** 6](#_Toc121289117)

# How to run

This program runs automated SQLi attacks on DVWA and records the network packets of the attack using a tcpdump.

## Move into VM

The program can be run from outside the virtual machine provided you have set up port forwarding from guest port 80 to host port 8080. In order to collect packets however the script must be run from inside the virtual machine.

The virtual machine must have TCPdump, python 3 and selenium installed. If you need to install any of these the commands are as such

sudo apt-get install tcpdump

sudo apt-get install python3

sudo python3 -m pip install selenium

Copy and paste the entire AutoSqliInject directory into your virtual machine. The documents  
folder is fine but anyone you can navigate to from the terminal is fine. Then Navigate to that folder in your terminal.

## Check User Settings

Open the UserSetUp.py document. Firstly chose whether or not you want to **HIDE\_BROWSER** or not. Then if you are running it in the VM, set **USING\_VM** to 1. Set your **SUDO\_PASS**. Mark whether or not you are **USING\_KALI**, Set your **USER\_LABEL** to your initials. All other settings can stay as the default. If you are using firefox make sure you set your **BROWSER\_CHOICE to 0.**

## Use Sudo

Don’t run the program as sudo. If a sudo command has not been run in the terminal already, the program will prompt you to enter your sudo password in order to run sudo start service apache and mysql.

## Python 3

To run the command simply type python3 AutoSQLI.py or python3 AutoComsI.py or even

python3 AutoBruteForce.py into the command line. The whole process should take around

two minutes to run.

# Breakdown of Files

## ExtModulesSetUp.py

This file just calls and installs all the external modules. Importing specific things from selenium, OS, Time, Random and RE.

## KaliSetUp.py

If you’re using Kali Linux make sure USING\_KALI is set to 1 in the UserSetUp.py, this file will activate the service mysql and apache2 it also doubles as a way of getting the user to enter the sudo command allowing the automated attacks to run unimpeached.

## LinkSetup.py

This sets up Errors, warnings, XPATHS of specific HTML elements, URLs, HTML Element Id’s and the default SQLI attacks

## UserSetUp.py

This is the file where you can make the most changes. Choose how the browser works, The log in credentials for the DVWA, The choice to use the VM or not, To generate sql attacks or use the default. To change the locations of the drivers for the browser. Change your sudo password. Change the labelling for the Pcaps. And change the number of iterations the Pcap files will do.

## SqliStatementGen

This file is in charge of automatically generating SQLI statements from the Poss\_SQLI\_Injections csv. The first method returns an array from a line in a CSV file that also contains extra comments in quotation marks. It is used on the Fourth line of the CSV file. The second splits the CSV file up into rows, picks a random value from each row and generates a random SQLI attack.

## **most\_common\_passwords.csv**

This is a csv file that contains 12000 of the most common passwords

## **most\_common\_passwords\_2.csv**

This is a csv file of 100 variations of each of the 12000 most common passwords

## Poss\_SQLI\_Injections.csv

This is a csv file that contains the constituents of an SQLI attack. The first row is just possible nonsense words. The second row just contains or. The third row is just variations of always true statements. The fourth row is injected commands and the fifth row is a comment.

## **Poss\_COMS\_Injections.csv**

This is a csv file that contains the constituents of an command injection attack. The first row is just possible nonsense words. The second row contains possible operators for low level security injections, the third row contains possible operators for medium level attakcs, and the fourth row operators for high level attacks. The last row is possible commands to be injected.

## BrowserSetUp.py

This file is where all the selenium commands are based. The first few modules are pretty basic commands which involve opening the browser, getting elements, sending information to specific elements. Sending information then pressing return. Waiting for an element to appear before carrying on. Execute a short java script insert into the HTML page. Clicking a link based on an XPAth. Jumping to another browser page. Logging in to a DVWA. Changing the security level. And the three attacks.

## BrowserSetUp.py

This file is where all the selenium commands are based. The first few modules are pretty basic commands which involve opening the browser, getting elements, sending information to specific elements. Sending information then pressing return. Waiting for an element to appear before carrying on. Execute a short java script insert into the HTML page. Clicking a link based on an XPAth. Jumping to another browser page. Logging in to a DVWA. Changing the security level. And the three attacks.

## Pcap\_Capture.py

These pcap captures run subprocesses in the terminal to capture and record all the pcaps. This will throw up a lot of text in the terminal and slow the process down.

## PassCracker.py

PassCracker is the file that reads in the list of 100 variations of each of the 12000 most popular passwords in USA. It reads the passwords into a dictionary for easy access.

## NameGenerator.py

The name generator just names a pcap file based on an array of text values. There is an absoluteNumber method aswell which returns the number of files in the SQLI attack folder so that the numbering can be absolute.

## AutoSQLI.py

This is the high level file you run to activate the whole script. It’s simple to run just navigate to the file, make sure you have all your set up in order and type **python3 AutoSQLI.py** it should take around 2 minutes to generate 20 Pcap files.

## **AutoComsI.py**

This is the high level file you run to activate the Automatic Command Injection scrtipt script. It’s simple to run just navigate to the file, make sure you have all your set up in order and type python3 AutoComsI.py it should take around 2 minutes to generate 20 Pcap files.

## **AutoBruteForce.py**

This is the high level file you run to activate the Automatic Brute force script. It’s simple to run just navigate to the file, make sure you have all your set up in order and type python3 AutoBruteForce.py it should take around 3 minutes to generate 20 Pcap files.

## /Drivers

A browser driver is what allows the browser to run, starts all the browsers scripts. You need to have the correct browser driver for your edition of the browser. This script assumes you are using google chrome or firefox and has the option for either. The Chrome Driver is v106.0.5249.61 and I think the geckodriver is version 0.32.0. A Browser driver probably exists somewhere in your system already in the application folder of your browser. But to save anyone having to go looking for it and because they are only about 10mb each the script has been directed it to automatically launch the browser from these by default.

/SQLIATTACKS

This is the folder where your pcap files will be sent after your tcp dump. If you use absolute numbering this file will fill up with many many pcap files. If you turn off absolute numbering, relative numbering will be used and the TCPdump will overwrite any old files with the same name.

## **/COMSIATTACKS**

This is the folder where your pcap files will be sent after your tcp dump from the command injection attack. If you use absolute numbering this file will fill up with many many pcap files. If you turn off absolute numbering, relative numbering will be used and the TCPdump will overwrite any old files with the same name.

## **/BRUTEFORCEATTACKS**

This is the folder where your pcap files will be sent after your tcp dump from the Brute Force Password attack. If you use absolute numbering this file will fill up with many many pcap files. If you turn off absolute numbering, relative numbering will be used and the TCPdump will overwrite any old files with the same name.